CLAIMS

What is claimed is:

1. A method, comprising:

building a steganographic extractor to extract hidden information contained in binary images;

loading the steganographic extractor during a pre-boot phase of a computer system;

discovering, during the pre-boot phase, binary firmware images on which a steganographic operation has been performed to generate hidden information contained within each binary firmware image; and

extracting, via the steganographic extractor during the pre-boot phase, the hidden information contained in each binary firmware image that is discovered.

2. The method of claim 1, further comprising:

obtaining a digital signature contained in the hidden information that is extracted from one of the binary firmware images;

comparing the digital signature with a known authentic digital signature to determine an authenticity of that binary firmware image.

3. The method of claim 2, further comprising:

loading the binary firmware image if is determined that the digital signature that is extracted matches the known authentic digital signature.

4. The method of claim 2, further comprising:

providing a notification message to at least one of a user and a system environment log indicating the binary firmware image could not be authenticated if it is found that the digital signature that is extracted does not match the known authentic digital signature.

5. The method of claim 4, further comprising:

providing the user with an option to load the binary firmware image that could not be authenticated; and

loading the binary firmware image if the user requests to do so; otherwise not loading the binary firmware image.

6. The method of claim 1, wherein the hidden information that is extracted contains at least one of manufacturer and versioning information, the method further comprising:

obtaining said at least one of manufacturer and versioning information from the hidden information; and

storing said at least one of manufacturer and versioning information in an asset management log.

7. The method of claim 1, further comprising:

discovering both binary firmware images having hidden information and binary firmware images not having hidden information during the pre-boot phase;

determine whether each of the binary firmware images is a trusted type or a non-trusted type; and

directly loading a binary firmware image if it is determined to be a trusted type.

8. The method of claim 7, wherein binary firmware images discovered in a boot firmware device are deemed to be trusted, while binary firmware images stored in firmware storage that is external to the boot firmware device are deemed to be non-trusted.

9. The method of claim 1, wherein the steganographic extractor includes a replacement map that maps equivalent sequences of op codes to one another, with one of the equivalent op code sequences assigned a "0" state and the other assigned a "1" state, the method further comprising:

parsing a binary firmware image to identify replaceable op code sequences defined by the replacement map;

for each replaceable op code sequence found,

determining whether it corresponds to a "0" or a "1" state; and

adding the "0" or "1" state to an extracted binary string that is used to hold the hidden information that is extracted.

10. The method of claim 9, wherein the binary firmware image is parsed backward, beginning with the end of the image.

11. The method of claim 9, wherein the extracted binary string contains encrypted information, the method further comprising decrypting the extracted binary string.

12. The method of claim 9, further comprising converting the extracted binary string into an alphanumeric form.

13. A method comprising:

building a steganographic op code sequence replacement map comprising respective pairs of equivalent op code sequences that may be substituted for one another;

augmenting a binary firmware driver having an original set of op code sequences by replacing portions of the original set of op code sequences with equivalent op code sequences in accordance with the steganographic op code sequence replacement map to embed hidden information embodied as a binary string in an augmented binary firmware driver;

storing the augmented binary firmware driver on a media;

storing, on a firmware storage device, a steganographic extractor configured to extract hidden information that is embedded in binary encodings using the steganographic op code sequence replacement map.

14. The method of claim 13, wherein the media the augmented binary firmware driver is stored on comprises an Option ROM (read only memory) contained in one of a peripheral device or add-on card.

15. The method of claim 13, wherein the media the augmented binary firmware driver is stored on comprises a network device configured to enable downloading of the augmented binary firmware driver.

16. The method of claim 13, wherein the firmware storage device is part of a computer system manufactured by an OEM (original equipment manufacturer) vendor, the method further comprising:

verifying proper operation of a computer system configuration that includes a computer system manufactured by the OEM and an original third party binary

firmware driver corresponding to a system component that is manufactured by a third party supplier;

augmenting the original third party binary firmware driver by replacing portions of its original set of op code sequences with equivalent op code sequences in accordance with the steganographic op code sequence replacement map to embed hidden information including a digital signature known to the OEM manufacturer;

storing the digital signature known to the OEM manufacturer on the computer system; and

providing firmware with the computer system that is configured to authenticate binary firmware drivers by extracting any hidden information contained therein via the steganographic extractor and verifying whether a digital signature is contained in said any hidden information that matches the digital signature known to the OEM manufacturer.

17. The method of claim 13, further comprising:

performing a hash on the binary firmware driver to obtain a hash digest; and

storing one of the hash digest or data derived from the hash digest in the hidden information.

18. The method of claim 17, further comprising:

encrypting the hash digest using an encryption key.

19. The method of claim 13, wherein the hidden information includes asset management information.

20. The method of claim 13, further comprising encrypting the hidden information.

21. A machine-readable medium to provide instructions, which when executed perform operations, including:

loading a steganographic extractor during a pre-boot phase of a computer system;

discovering, during the pre-boot phase, binary firmware images on which a steganographic operation has been performed to generate hidden information embedded within each binary firmware image; and

extracting, via the steganographic extractor during the pre-boot phase, the hidden information contained in each binary firmware image that is discovered.

22. The machine-readable medium of claim 21, wherein the media further includes instructions embodied as the steganographic extractor.

23. The machine-readable medium of claim 21, further comprising instructions for performing the operations of:

obtaining a digital signature contained in the hidden information that is extracted from one of the binary firmware images;

comparing the digital signature with a known authentic digital signature to determine an authenticity of the binary firmware image.

24. The machine-readable medium of claim 21, further comprising instructions for performing the operations of:

determining op code sequences that are identifiable to the steganographic extractor as representing steganographic data;

performing a hash on a portion of a binary firmware image that includes op codes that are exclusive of the op code sequences that represent steganographic data to obtain an image hash digest;

extracting an authenticated hash digest from the hidden information; and

comparing the image hash digest to the authenticated hash digest to determine an authenticity of the binary firmware image.

25. The machine-readable medium of claim 21, further comprising instructions for performing the operations of:

retrieving a decryption key; and

decrypting the hidden information that is extracted with the decryption key to obtain the authenticated hash digest.

26. The machine-readable medium of claim 21, wherein the hidden information includes asset management information, and the machine-readable medium further includes instructions for performing the operations of:

retrieving the asset management information from the hidden information that is extracted; and

storing the asset management information that is retrieved.

27. A system comprising:

a processor;

a memory, coupled to the processor;

a flash device, coupled to the processor and having firmware instructions stored therein, which when executed perform operations including:

loading a steganographic extractor during a pre-boot phase of a computer system into the memory;

discovering, during the pre-boot phase, binary firmware drivers containing hidden information that was embedded via a steganographic operation; and

extracting, via the steganographic extractor during the pre-boot phase, the hidden information contained in each binary firmware image that is discovered.

28. The system of claim 27, wherein the flash device further includes firmware instructions embodied as the steganographic extractor.

29. The system of claim 27, wherein the flash device further includes firmware instructions for performing the operations of:

determining op code sequences that are mapped to identify steganographic data;

performing a hash on a portion of a binary firmware driver image that includes op codes that are exclusive of the op code sequences that are mapped to identify steganographic data to obtain an image hash digest;

extracting an authenticated hash digest from the hidden information; and

comparing the image hash digest to the authenticated hash digest to determine an authenticity of the binary firmware image.

30. The system of claim 29, wherein the hidden information was encrypted using a private key and wherein the flash device further includes firmware instructions for performing the operations of:

retrieving a public key stored in the computer system corresponding to the private key; and

6 employing the public key to decrypt the hidden information that is extracted
7 from one of the binary firmware drivers.